New Architectures for PUF Authentication and Key Generation

Mandel Yu, Chief Scientist

Collaborators and Acknowledgements:

Prof. Srinivas Devadas, MIT, Verayo co-founder & CTO

Prof. Ingrid Verbauwhede, KU Leuven COSIC Crypto Group

Matthias Hiller, Technical University Munich / Institute for Security in Information Technology

Verayo Engineering Staff

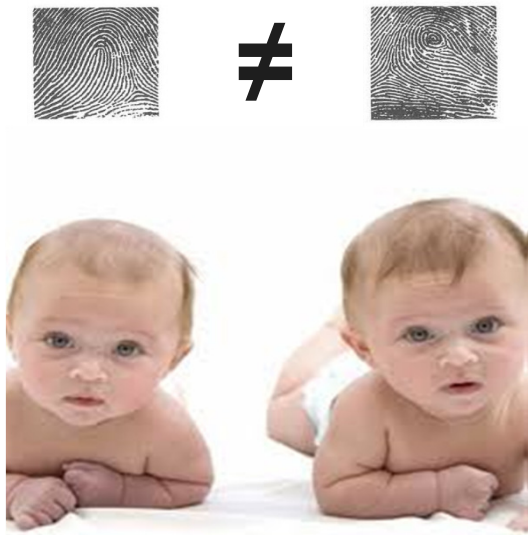
Agenda

- Introduction
  - *Silicon "Biometric"*
  - *Use Cases: Authentication, Key Generation*
- PUF Authentication: Noise-Bifurcation
  - *Adversary & Verifier Sees Asymmetric Noise*
  - *Adversary has Imperfect CRPs to Launch Machine Learning Attack*
- PUF Key Generation: Optimal Symbol Decoding
  - *Provably Optimal Receiver*
  - *Orders of Magnitude Better Results than current BCH, Rep Coding…*
- Conclusions

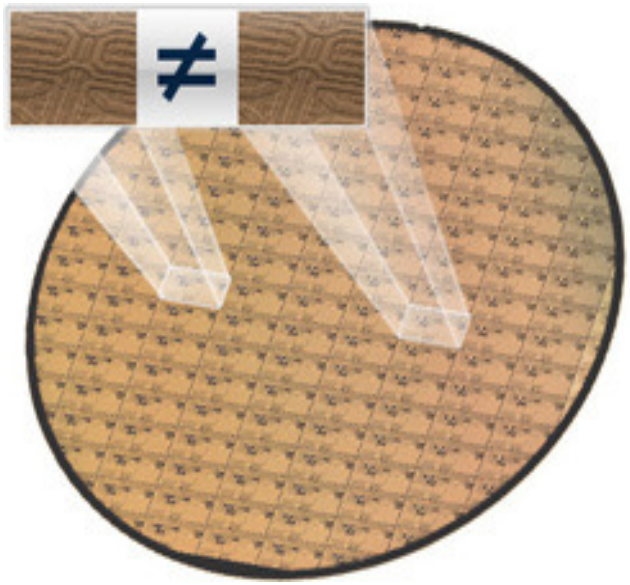
Introduction

Silicon “Biometric” Technology

Physical Unclonable Functions (PUFs)



Each of us is unique and different…



… and so are silicon chips

Two Main PUF Use Cases

PUF Authentication

Use PUF as a function: challenge → response

Only an authentic IC can produce a correct response for a challenge

Inexpensive: no special fabrication/masking steps, no key/crypto needed

40M+ units shipped for anti-counterfeiting applications

PUF Key Generation

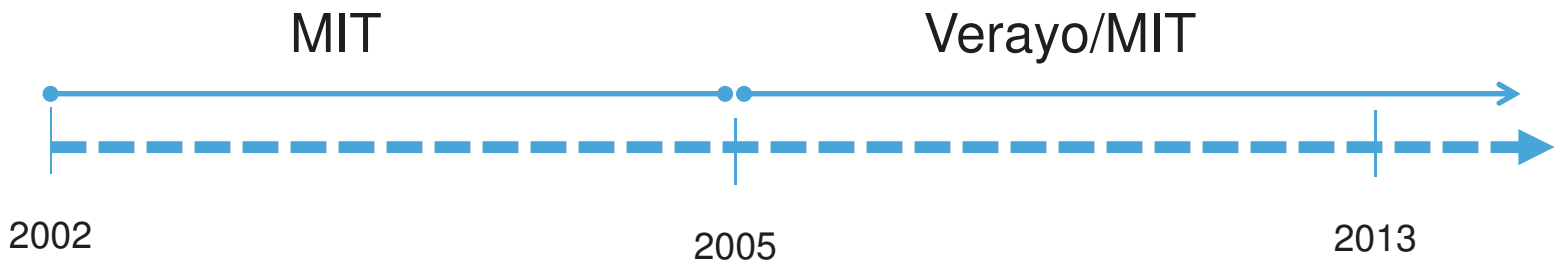
Highly secure: volatile secrets, no need for trusted programming

Can integrate key generation into an IC or an FPGA

Key can be used for cryptographic applications (AES, RSA, HMAC…)

Customer engagements: 0.13u, 90nm, 65nm, 45nm, 28nm and below (FPGA, ASIC)

Timeline



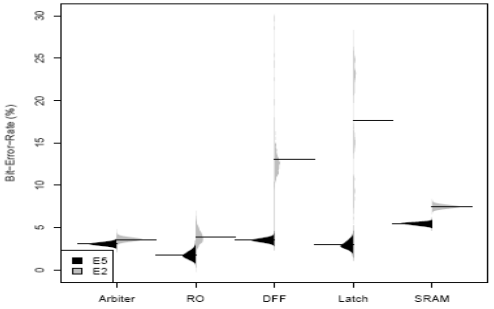
Concept & Prototype

- “Silicon Physical Random Functions”, 2002
- First PUF on custom silicon, 2004

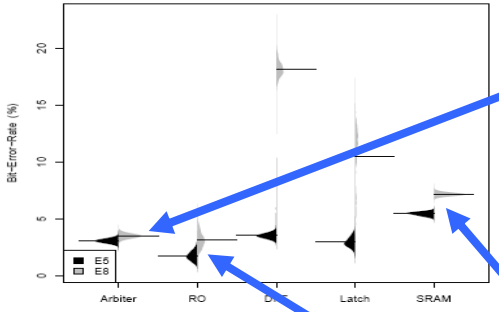
Improvements & Commercialization

- Stability-enhanced PUF, 2008
- Supply chain TRUST FPGA, 2009
- MIL-SPEC Temperature, VT Corners, Aging, NIST tests, Radiation, 2010
- 1M+ chips shipped, 2011
- 40M+ chips shipped, 2014

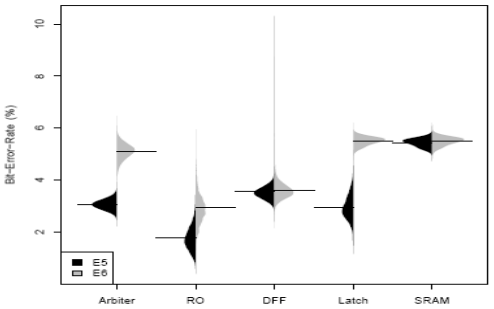
Comparisons: PUF Noise



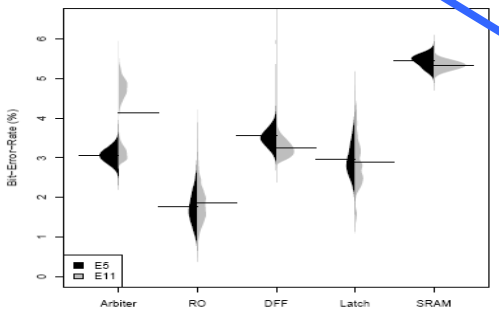
(a) Bit error rates at +25 °C (test case E_5, black) and at −40 °C (test case E_2, gray)



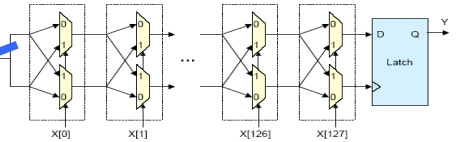
(b) Bit error rates at +25 °C (test case E_5, black) and at +85 °C (test case E_8, gray)

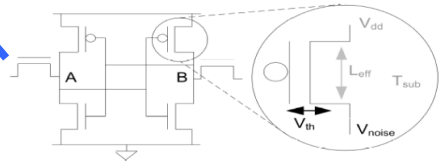


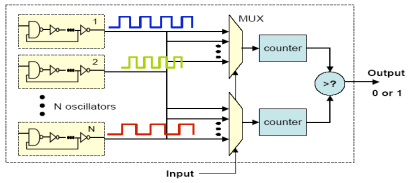
(c) Bit error rates at 1.20 V (nominal supply voltage, test case E_5, black) and at 1.32 V (+10% overvoltage,



(d) Bit error rates with active core off (test case E_5, black) and active core on (test case E_{11}, gray)



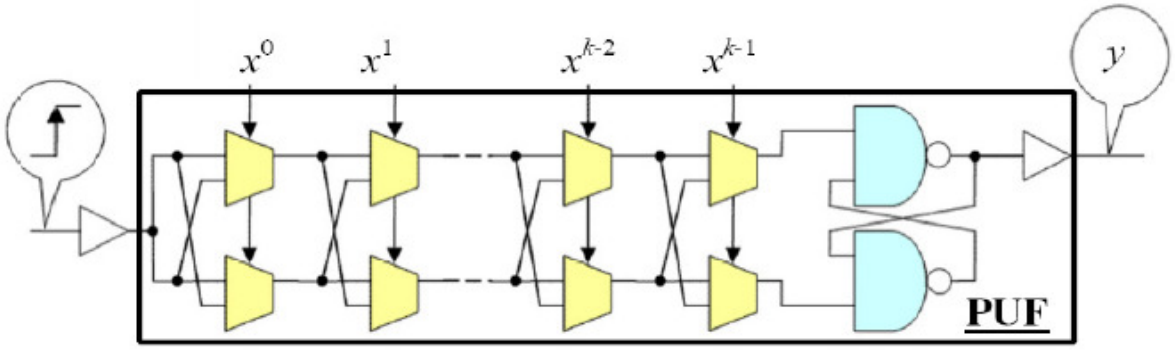




From: Katzenbeisser, Kocabas, Rozic, Sadeghi, Verbauwhede, Wachsmann, CHES 2012

PUF Authentication: Noise Bifurcation

Detecting Manufacturing Variation Reliably, Quickly



(“Basic” PUF Building Block, XORs not shown)

Circuit balanced by design

Allows manufacturing variation (MV) to show through

Differential measurements

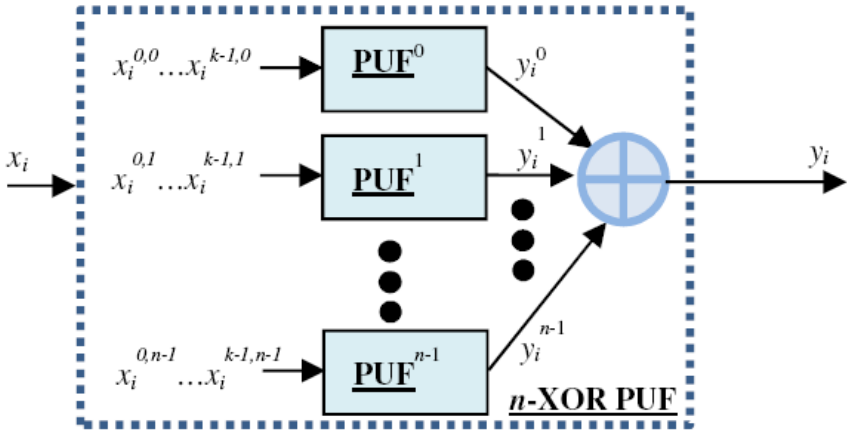
Cancel voltage, temperature, aging effects that overwhelm MV

But physical function building blocks are linear…

Prevents MV noise explosion, but is also easy to model → ADD NON-LINEARITY

From: Gassend, Clarke, Dijk, Devadas, CCS 2002

n-XOR Security Scaling



Bit Length	Pred. Rate	No. of XORs	CRPs ($\times 10^3$)	Training Time
64	99%	4	12	3:42 min
		5	80	2:08 hrs
		6	200	31:01 hrs

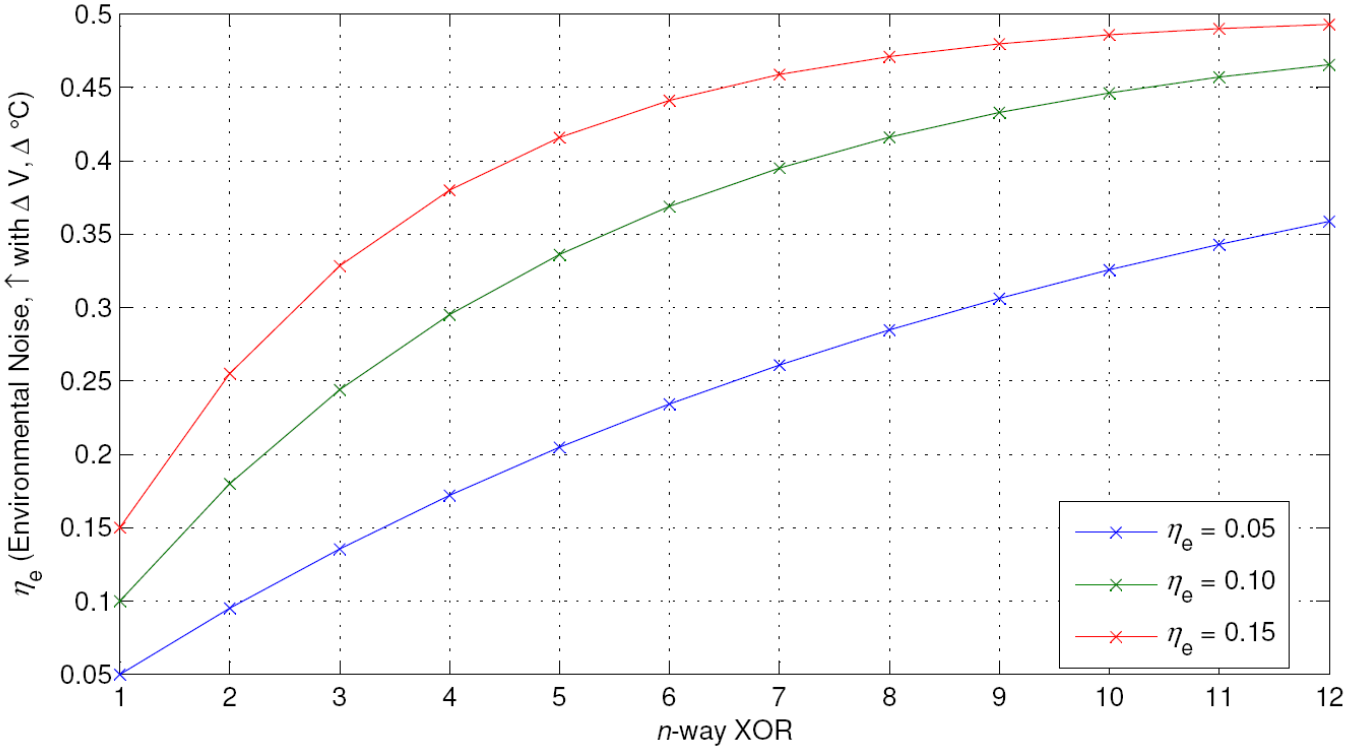
From: Rührmair, Sölter, Sehnke, Xu, Mahmoud, Stoyanova, Dror, Schmidhuber, Burleson, Devadas, IEEE TIFS, 2013.

XORs ↑ Machine Learning (ML) Attack Complexity

8 XORs, 12 XORs, 16 XORs…

More Training Time, Challenge/Response Pairs (CRPs)

Side-Effect: Increases Authentication Noise



XORs ↑ Environmental Noise Towards 50% "Wall"

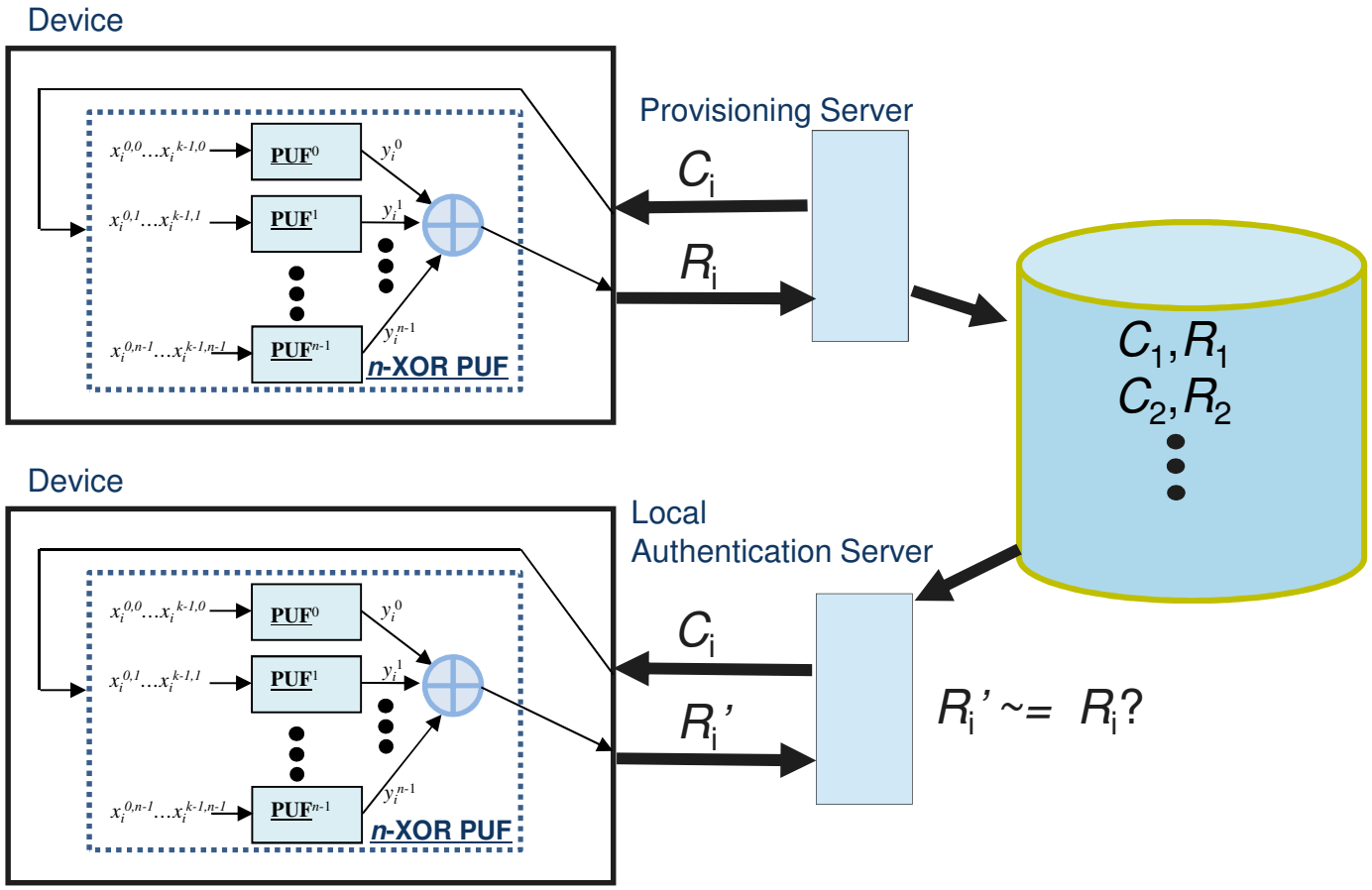
At 50%, Adversary Cannot Distinguish PUF from Random

… *but Verifier Cannot Authenticate Either*…

CRP-Based Authentication

Provisioning:
$R_i = \text{PUF}(C_i)$

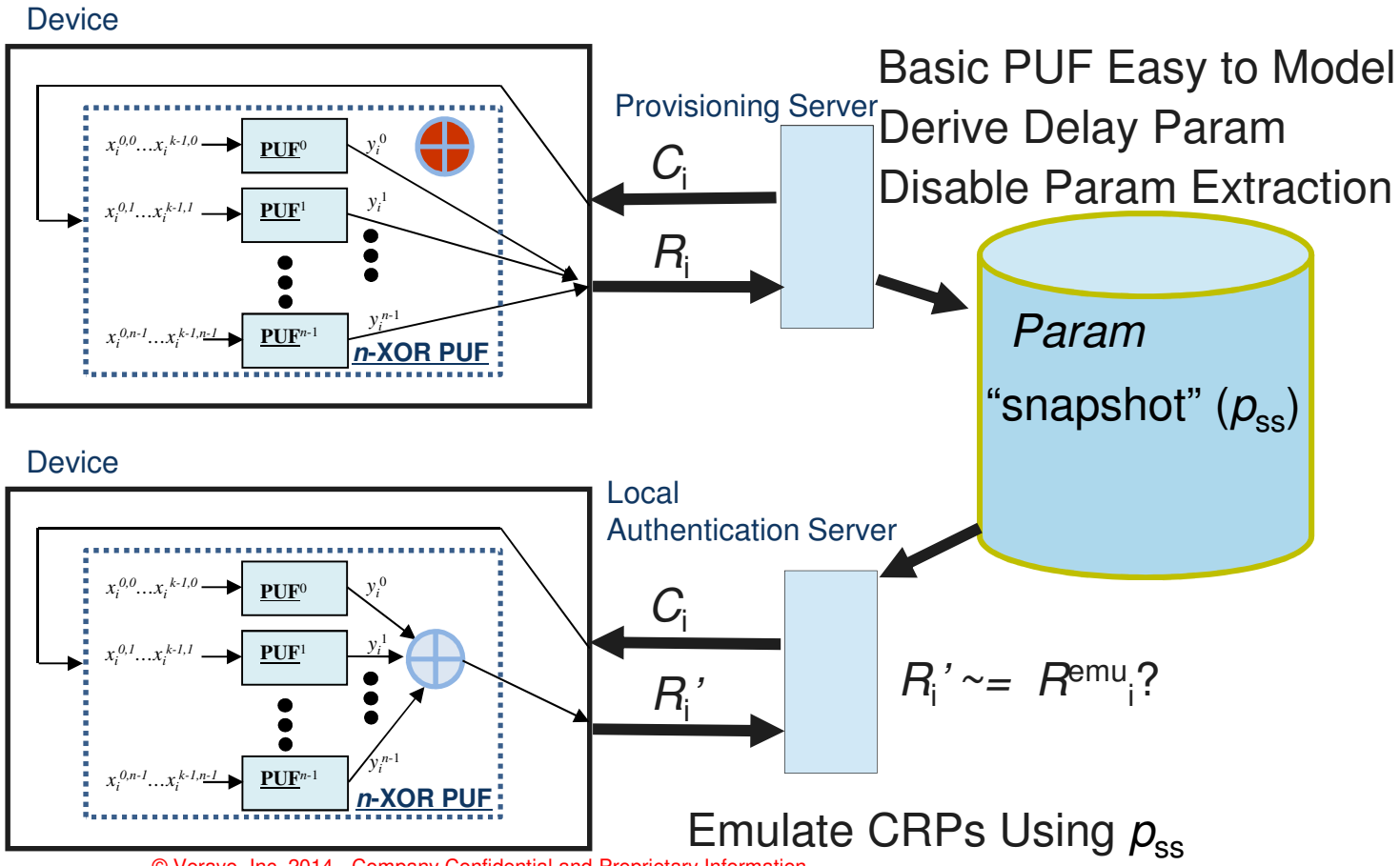
Authentication:
$R_i' = \text{PUF}(C_i)$



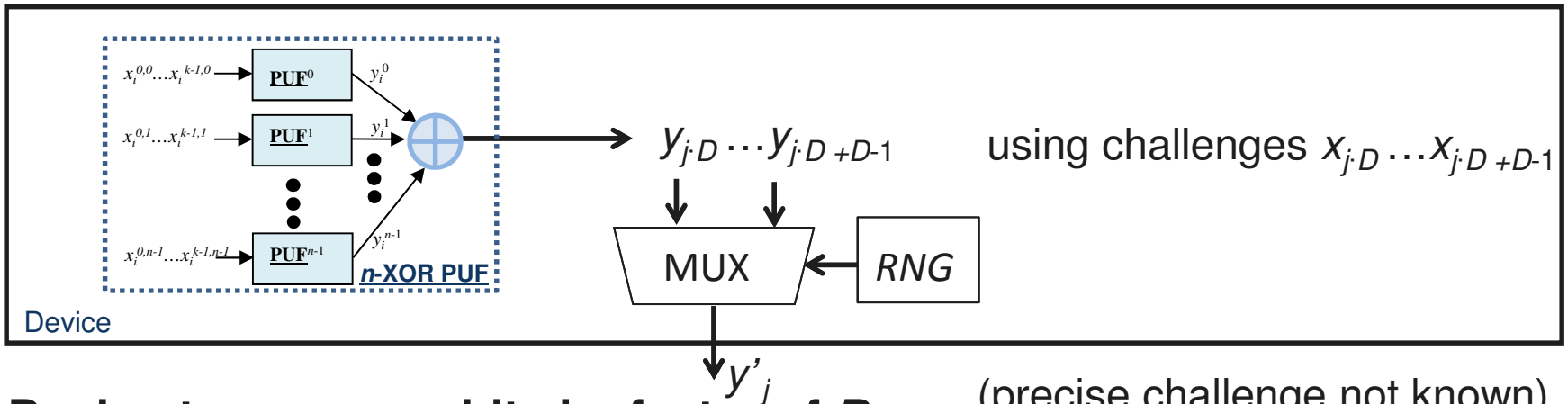
Parameter-Based Authentication

Provisioning:
$R_i = \text{PUF}(C_i)$

Authentication:
$R_i' = \text{PUF}(C_i)$



Noise Bifurcation via Randomized Decimation



Decimate response bits by factor of *D*:

(precise challenge not known)

Randomly output 1:*D* response bits

Adversary doesn't know *precise* challenge

CRP "noise" increases with *D*

Verifier can still authenticate using "extra information" p_{ss}

Authenticate on bit locations where *pre-decimated response choices* are all 1s or all 0s (no ambiguity in y'_j)

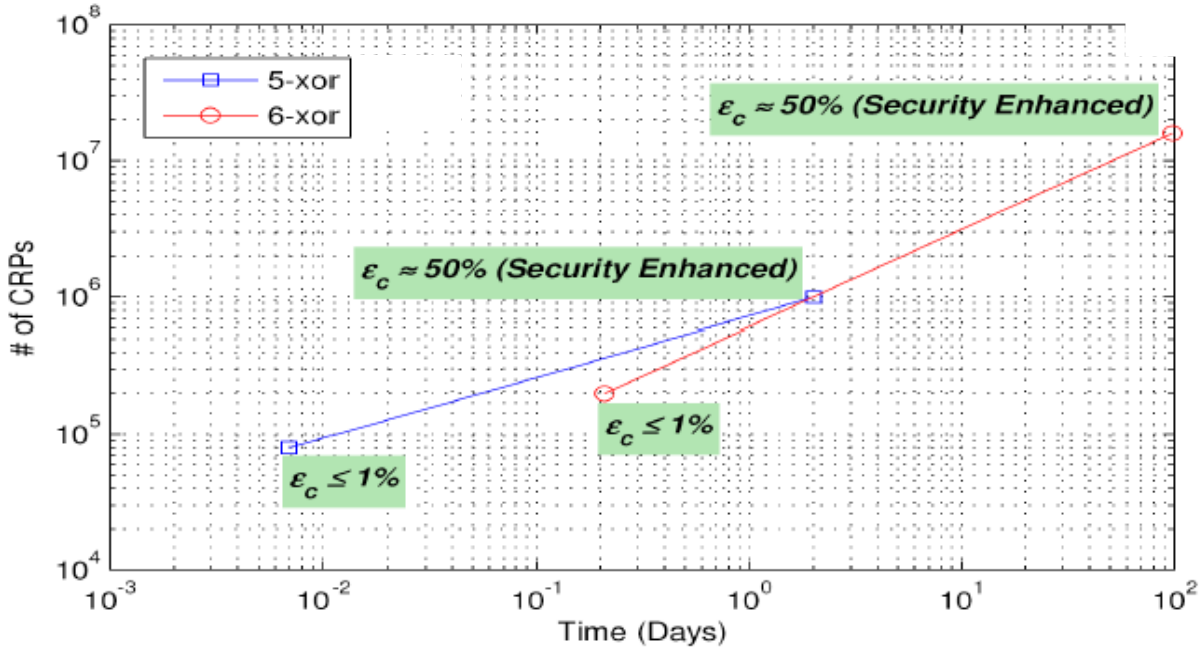
What Adversary Sees

Adversary <u>lacks clean CRPs</u> to launch ML attack

Can't replay challenges to filter out CRP noise (due to challenge seeds exchange)

Need to make *imperfect inferences* to form CRP training labels

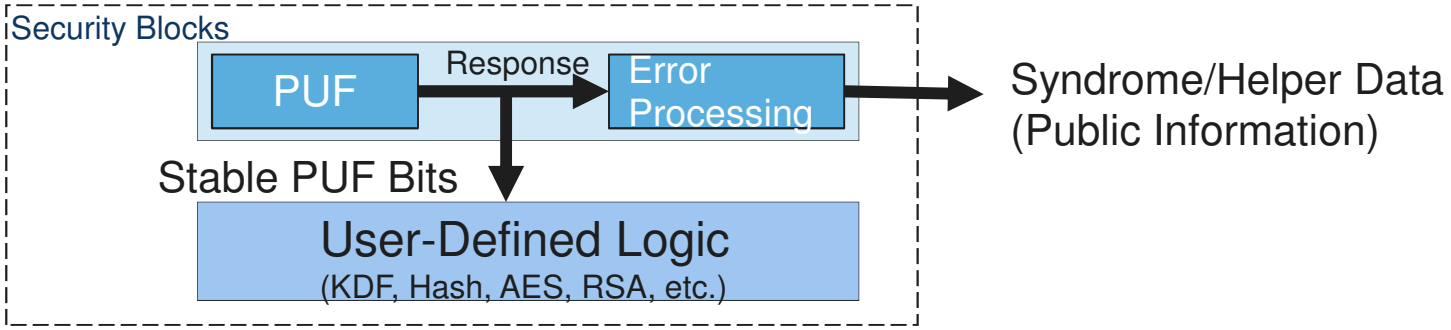
ML attack effort ↑ 10x or 100x or more at a mere $D = 2$



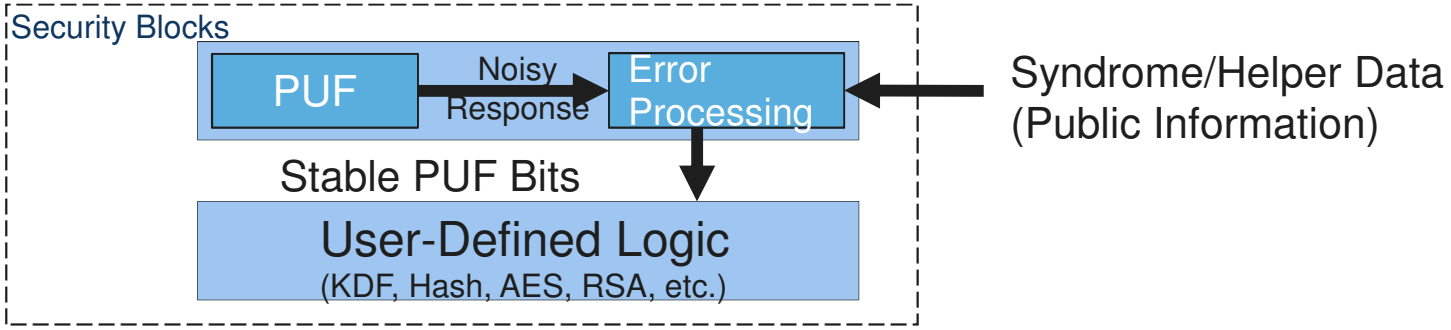
PUF Key Generation: Optimal Symbol Decoding

PUF Key Generation Steps

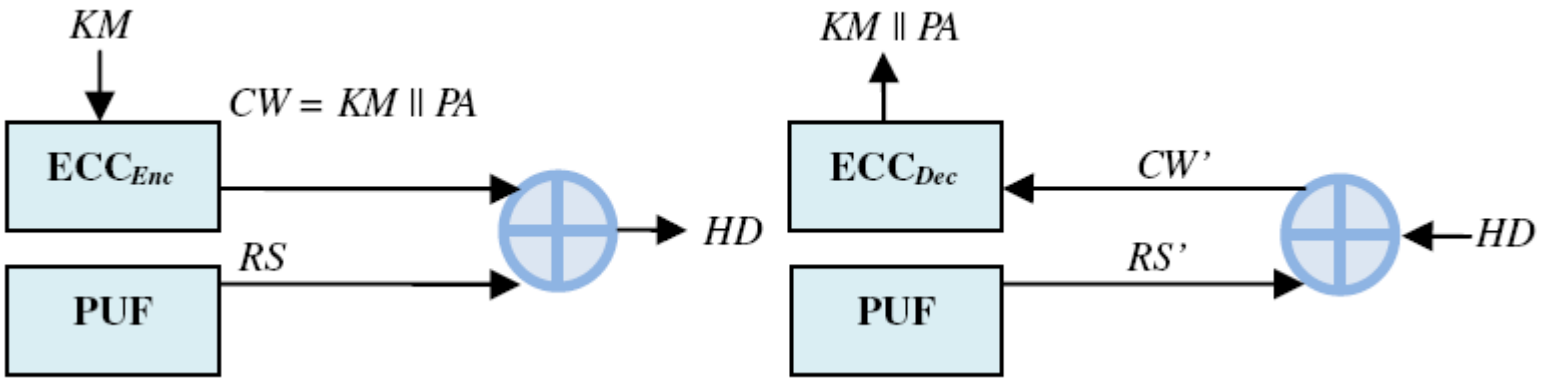
Provisioning:



Regeneration:



Code-Offset Helper Data (Conventional)



Code-Offset Helper Data:

HD relates *RS* (PUF response) to (codeword) *CW*

RS now in a form where conventional error correction can be applied

Published Approaches Use *Binary* Codewords:

Each "letter" (position) in codeword a single bit

$P(X) = m_0 + m_1X + \ldots + m_{k-1}X^{k-1}$, where $m_0 \ldots m_{k-1} \in GF(2) = \{0,1\}$

Symbol-Based Decoding (New)

<u>*Non-Binary*</u> Symbols:

Each codeword position a "letter" (symbol) vs. a single bit:

$P(X) = m_0 + m_1X + \ldots + m_{k-1}X^{k-1}, \qquad m_0, \ldots, m_{k-1} \in GF(2) = \{0,1\}$

$P(X) = m_0 + m_1X + \ldots + m_{k-1}X^{k-1}, \qquad m_0, \ldots, m_{k-1} \in GF(2^k) = \{0,1, \ldots, 2^{k-1}\}$

Benefits:

<u>*Orders of magnitude*</u> improved performance vs. current BCH, repetition coding…
holding PUF noise, helper data size constant

Natural soft-decision metric (based on symbol distances)

Natural code concatenation mechanism (similar to symbol-based Reed Solomon)

Reduced Second Stage ECC Complexity (due to high first stage gain)

Resistance to helper data manipulation attacks

Three-Pronged Validation

Mathematics

Derived Provably Optimal Symbol Decoder

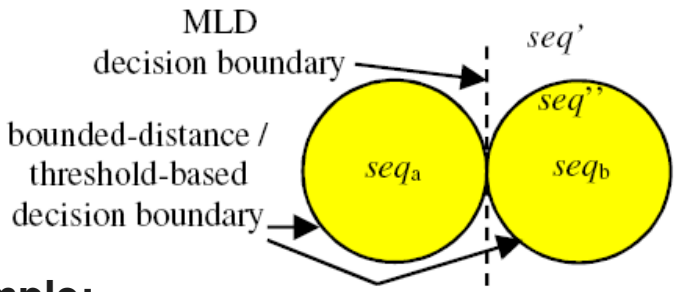
Simulation

Better Results than BCH, rep coding

Empirical

Xilinx 28nm Artix-7 Implementation, MIL-SPEC temperature

Maximum Likelihood Symbol Detection (Provably Optimal)



Example:

seq_a = 1111111

seq_b = 1111000

seq' = 0000000

Conventional ECC:

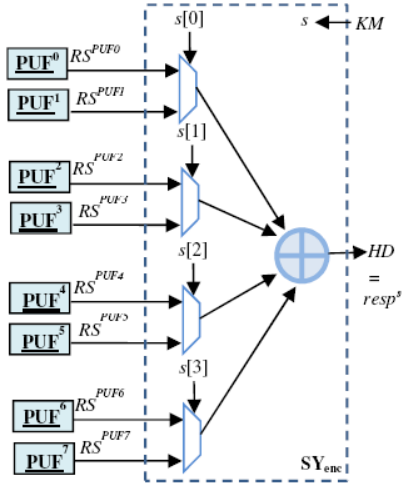
seq' outside hamming sphere of seq_b

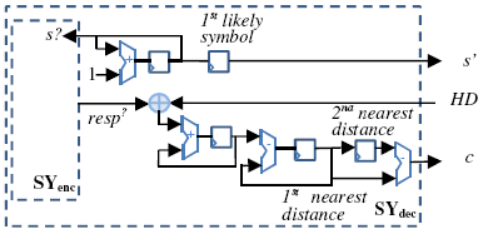
decodes incorrectly (≥ half bits different!) ☹

Maximum Likelihood:

seq' closer to seq_b than to seq_a

decodes correctly ☺



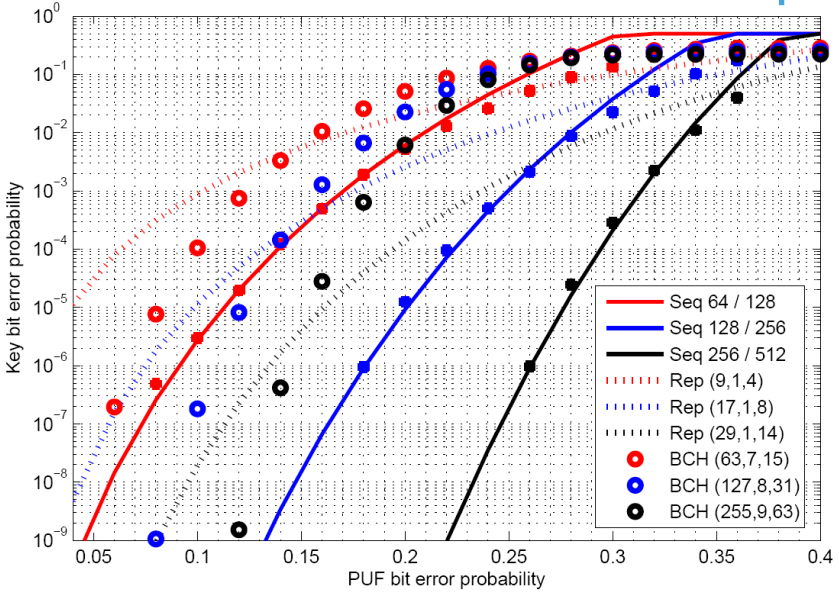


4-bit Symbol Codec:

~15 SLICES, 440 nand2 equiv, serial impl.

"Noise-Bifurcated" Helper Data

Matlab Simulation and Empirical Emulation



ENVIRONMENTAL STRESS TESTING, 28*n*M FPGA SILICON

Prov	Regen	Key Regen Cycles	After MLD	After SD
25°C	-65°C	$3.4x10^5$	11 Failures ($3.24x10^{-5}$)	0 Failures ($<<$ 2.94ppm)
25°C	105°C	$1.2x10^6$	3562 Failures ($2.95x10^{-3}$)	0 Failures ($<$ 0.83ppm)
1.00V	1.05V	$1.7x10^6$	0 Failures	0 Failures ($<<$ 0.59ppm)
1.00V	0.95V	$8.0x10^5$	0 Failures	0 Failures ($<<$ 1.25ppm)

Xilinx Artix-7 Emulation:

Error free performance: 1M+ key regen cycles

Includes MIL-SPEC temperatures

Analytical and Monte Carlo Simulation:

@ 0.15 PUF noise (x-axis), symbol sequence length of 128 bits (solid blue curve) has bit error rate (y-axis) of ~$1x10^{-8}$ (*before* any soft-decision decode) *BCH, rep 10,000x worse*

Actual PUF noise level depends on env. condition, process geometry, particular PUF circuit impl.

Increase sequence length to scale for higher PUF noise (better noise scaling than BCH, rep)

Linear-XOR Malleability

Code-offset Helper Data is Linearly-Malleable (Conventional)

XOR of helper data with another codeword causes a differential shift in the codeword processed inside device, resulting in an attack From: K. Karakoyunlu, B. Sunar, WIFS 2010

Device-Specific Symbol Generation (New)

Symbol-to-symbol relationship depend on manufacturing variations

vs. an algorithmic codeword structure

Helps to address XOR malleability and other similar helper data linear-modification attacks

Conclusions

Conclusions

Noise-Bifurcated Authentication:

- Improved Security Scaling (vs. *n*-XOR)
- Verifer's Noise η_v Constant
- Adversary's Noise $\eta_a \rightarrow 50\%$ where PUF is Indistinguishable From Random

Robust Symbol-Based Key Generation:

- Temperature: -65°C to 105°C, ambient (includes 125°C junction)
- Voltage: 0.95V to 1.05V
- "Noise-Bifurcated", Not Linearly-Malleable Helper Data

3-Pronged Validation:

- Mathematics: Provably Optimal Symbol Decoding
- Simulation: Better Results than current BCH, repetition coding…
- Emulation: Xilinx Artix-7 results with error rates ≤ 1ppm

Thank you.